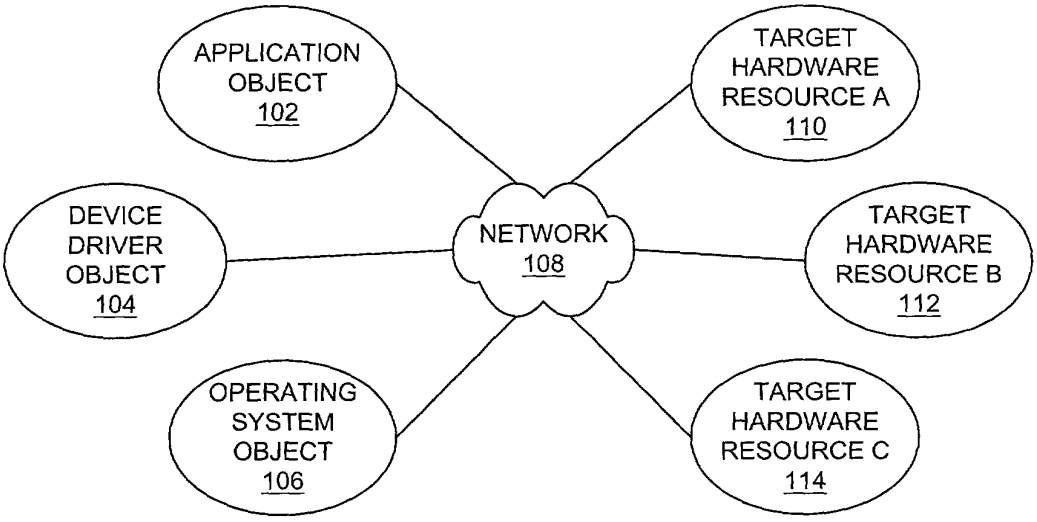

FIG. 1
(PRIOR ART)

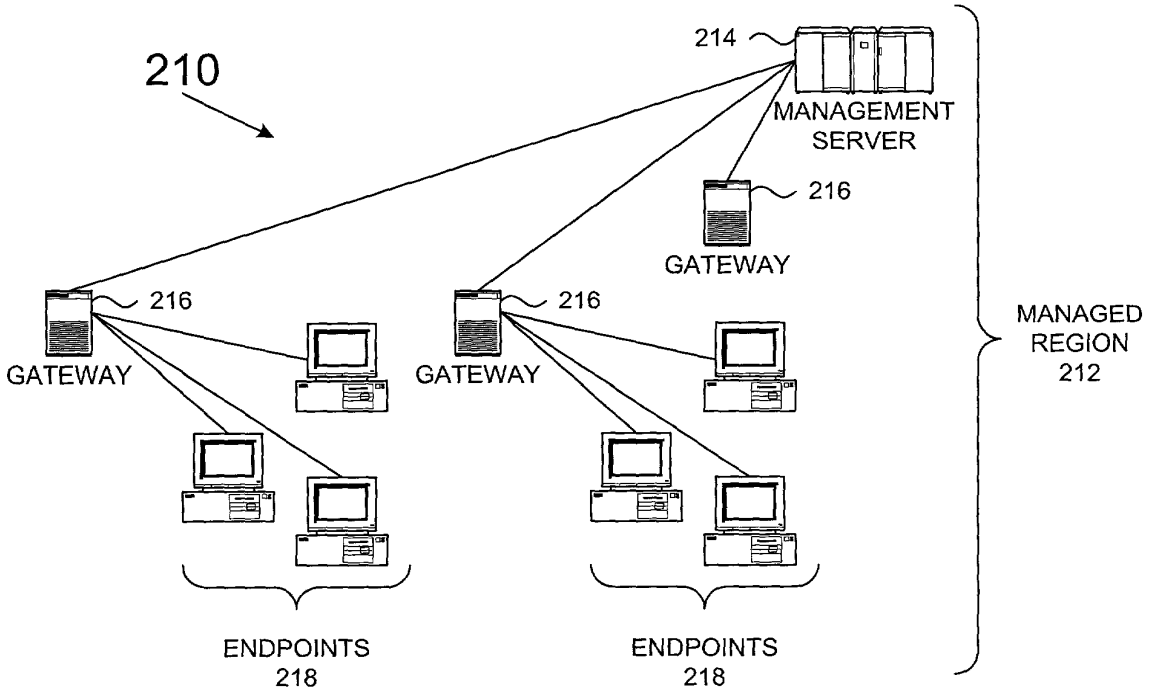

FIG. 2A

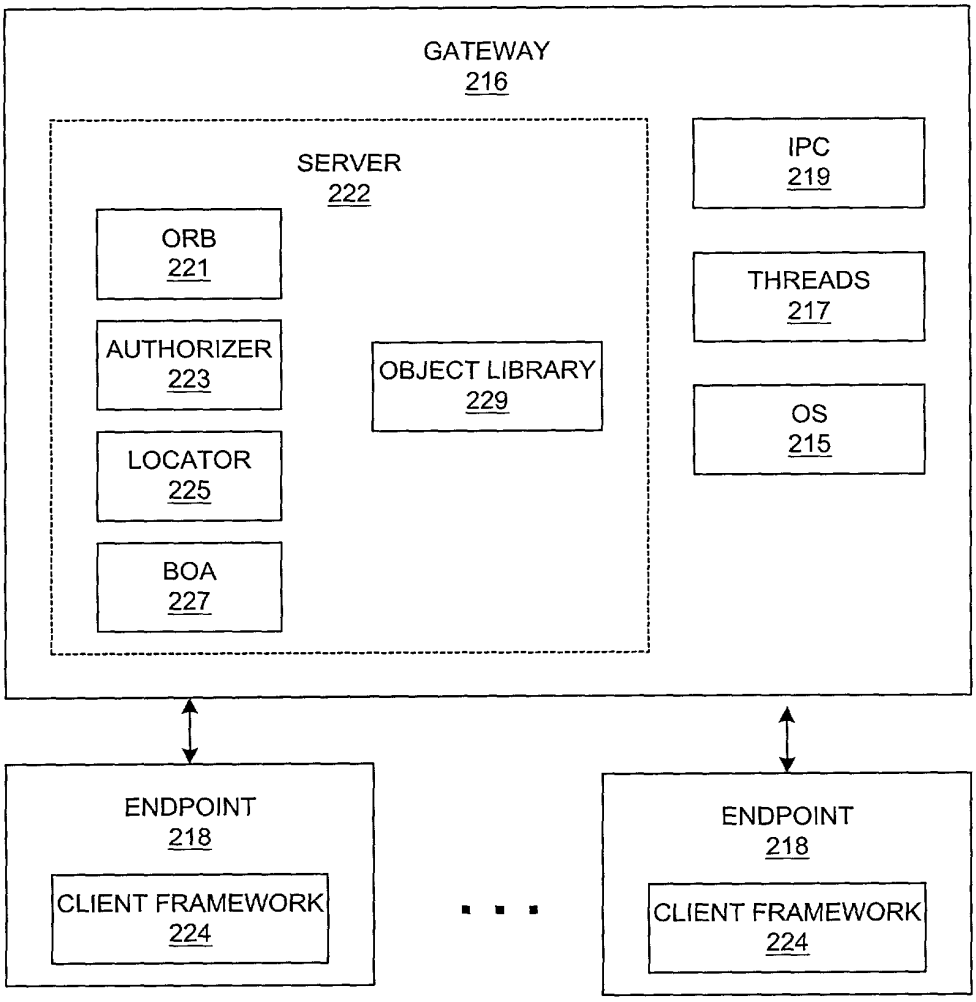
GATEWAY
216
SERVER
222
ORB
221
AUTHORIZER
223
LOCATOR
225
BOA
227
OBJECT LIBRARY
229
IPC
219
THREADS
217
OS
215
ENDPOINT
218
CLIENT FRAMEWORK
224
ENDPOINT
218
CLIENT FRAMEWORK
224

FIG. 2B

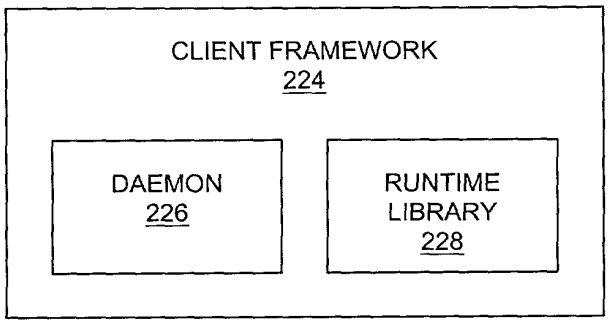
CLIENT FRAMEWORK
224
DAEMON
226
RUNTIME
LIBRARY
228

FIG. 2C

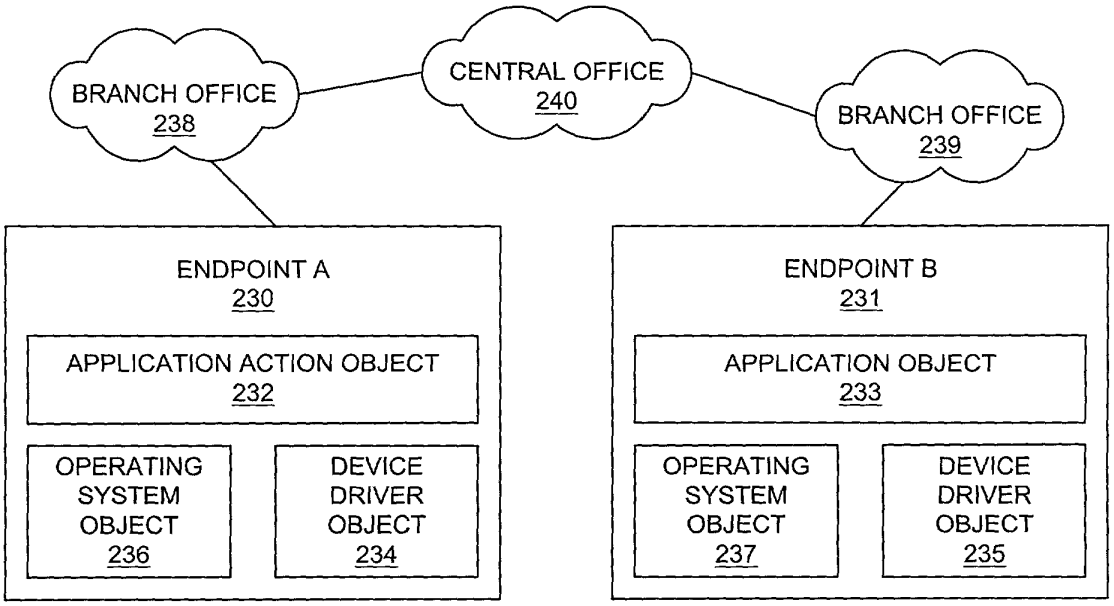
BRANCH OFFICE
238
CENTRAL OFFICE
240
BRANCH OFFICE
239
ENDPOINT A
230
APPLICATION ACTION OBJECT
232
OPERATING SYSTEM OBJECT
236
DEVICE DRIVER OBJECT
234
ENDPOINT B
231
APPLICATION OBJECT
233
OPERATING SYSTEM OBJECT
237
DEVICE DRIVER OBJECT
235

FIG. 2D

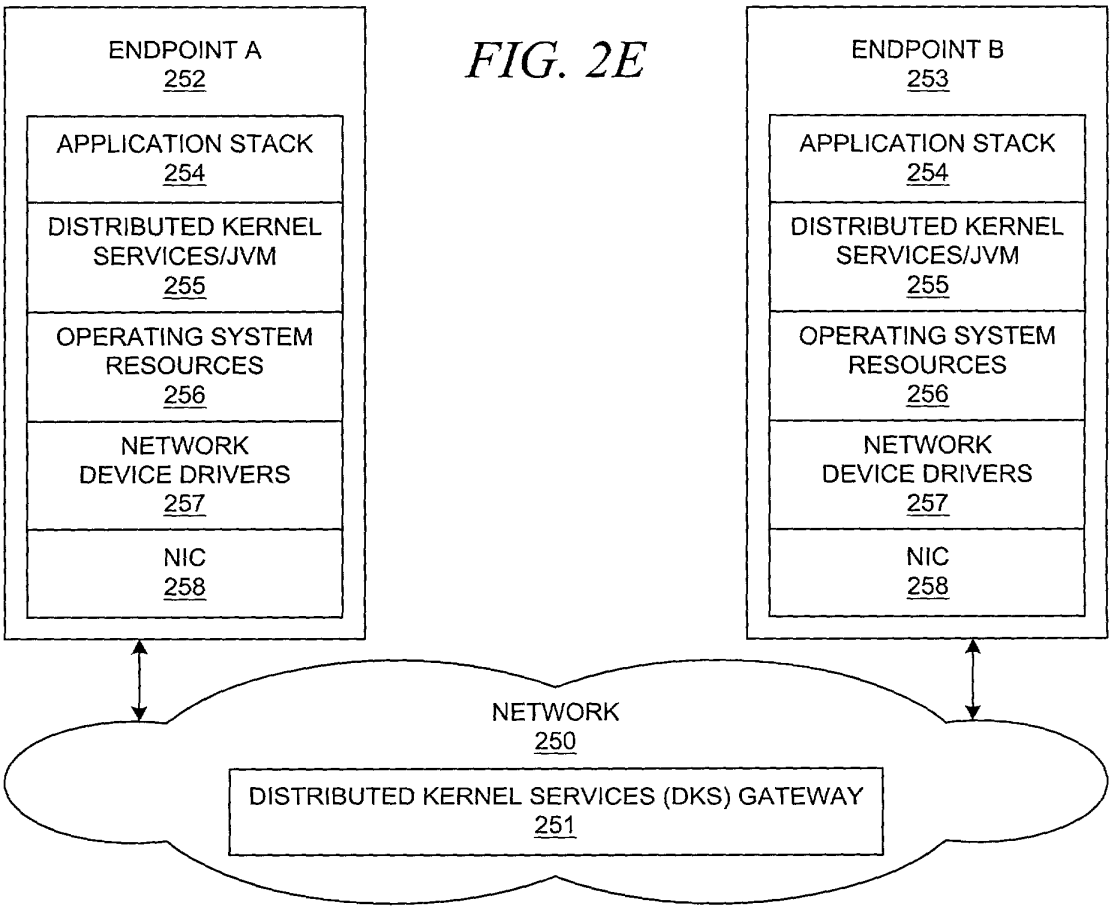
ENDPOINT A
252
APPLICATION STACK
254
DISTRIBUTED KERNEL SERVICES/JVM
255
OPERATING SYSTEM RESOURCES
256
NETWORK DEVICE DRIVERS
257
NIC
258
ENDPOINT B
253
APPLICATION STACK
254
DISTRIBUTED KERNEL SERVICES/JVM
255
OPERATING SYSTEM RESOURCES
256
NETWORK DEVICE DRIVERS
257
NIC
258
NETWORK
250
DISTRIBUTED KERNEL SERVICES (DKS) GATEWAY
251

FIG. 2E

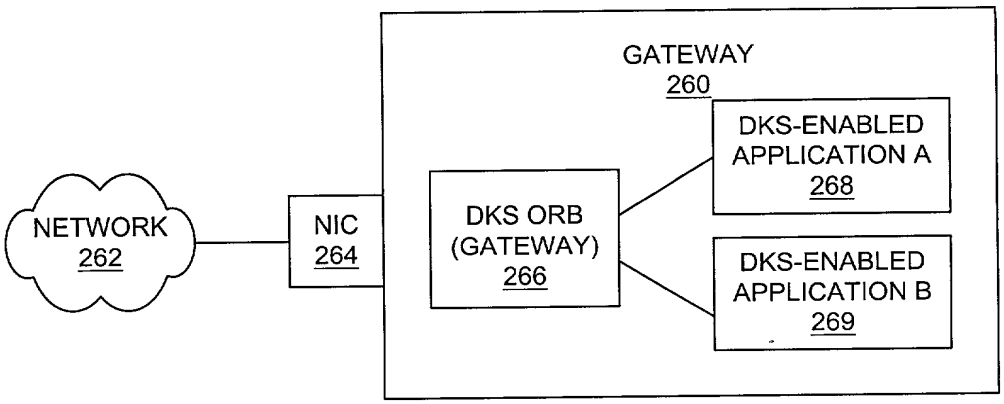
GATEWAY
260
DKS-ENABLED
APPLICATION A
268
NETWORK
262
NIC
264
DKS ORB
(GATEWAY)
266
DKS-ENABLED
APPLICATION B
269

FIG. 2F

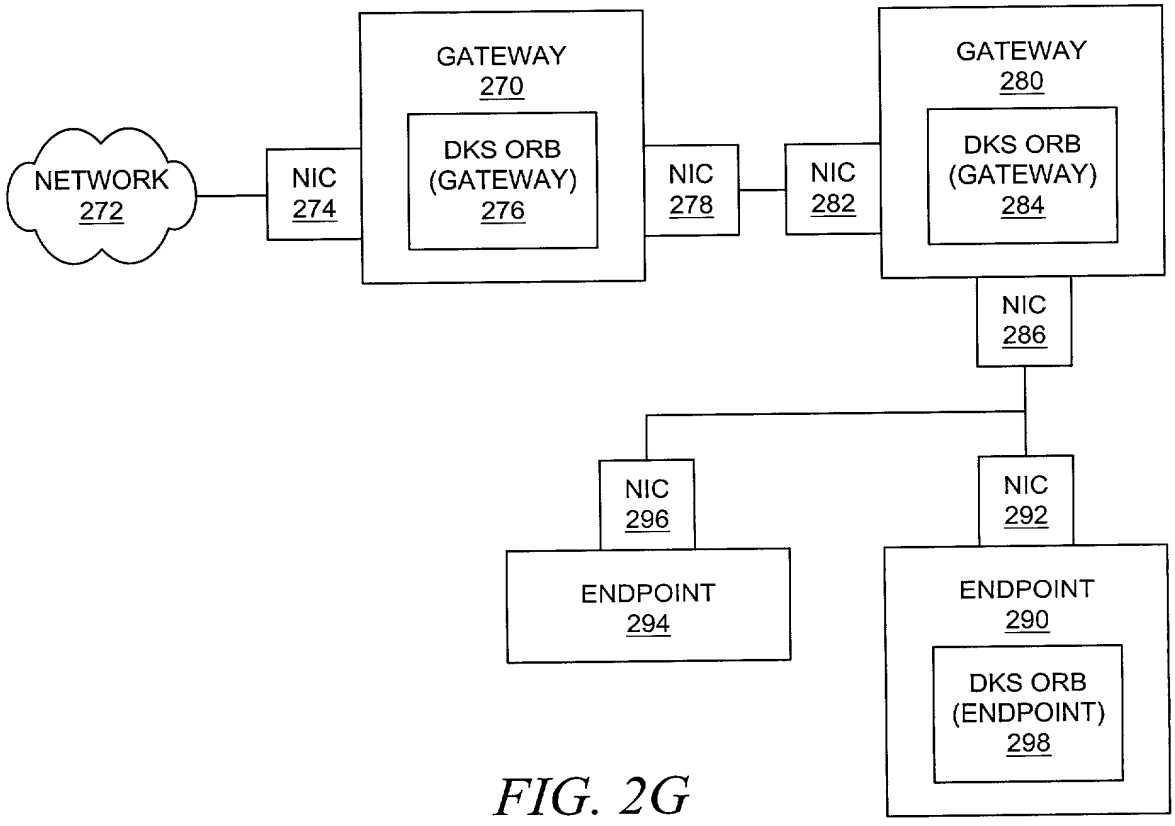
GATEWAY
270
GATEWAY
280
NETWORK
272
NIC
274
DKS ORB
(GATEWAY)
276
NIC
278
NIC
282
DKS ORB
(GATEWAY)
284
NIC
286
NIC
296
NIC
292
ENDPOINT
294
ENDPOINT
290
DKS ORB
(ENDPOINT)
298

FIG. 2G

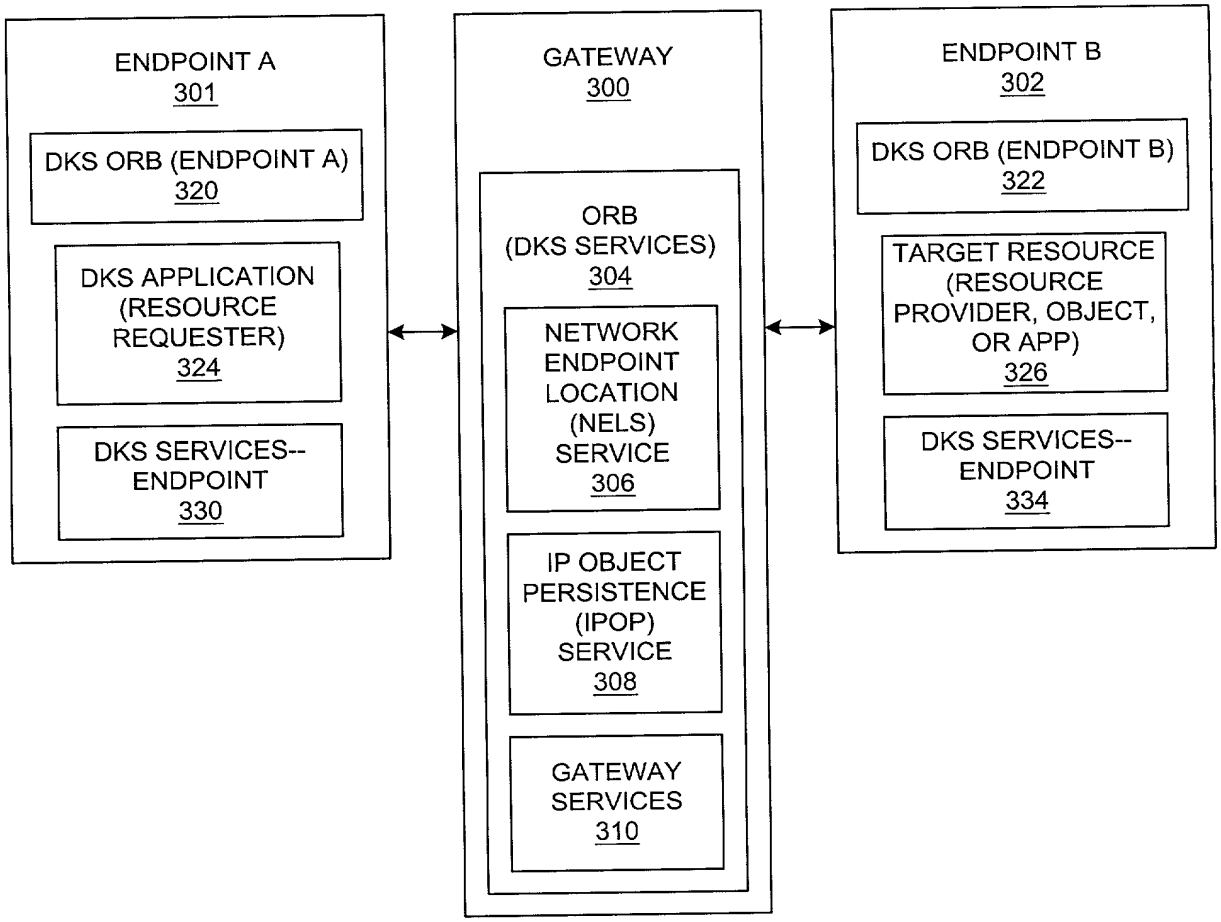
ENDPOINT A
301
DKS ORB (ENDPOINT A)
320
DKS APPLICATION
(RESOURCE
REQUESTER)
324
DKS SERVICES--
ENDPOINT
330
GATEWAY
300
ORB
(DKS SERVICES)
304
NETWORK
ENDPOINT
LOCATION
(NELS)
SERVICE
306
IP OBJECT
PERSISTENCE
(IPOP)
SERVICE
308
GATEWAY
SERVICES
310
ENDPOINT B
302
DKS ORB (ENDPOINT B)
322
TARGET RESOURCE
(RESOURCE
PROVIDER, OBJECT,
OR APP)
326
DKS SERVICES--
ENDPOINT
334

FIG. 3

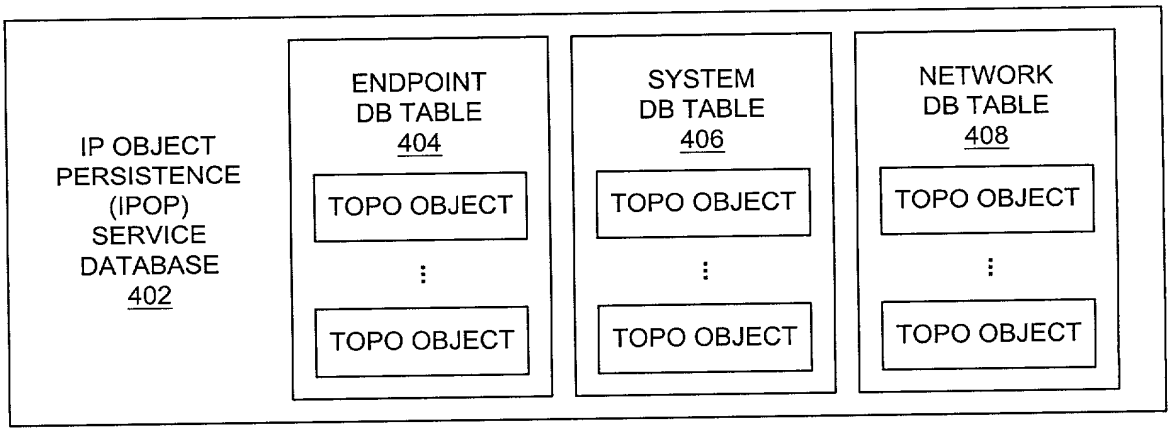
IP OBJECT
PERSISTENCE
(IPOP)
SERVICE
DATABASE
402
ENDPOINT
DB TABLE
404
TOPO OBJECT
TOPO OBJECT
SYSTEM
DB TABLE
406
TOPO OBJECT
TOPO OBJECT
NETWORK
DB TABLE
408
TOPO OBJECT
TOPO OBJECT

FIG. 4

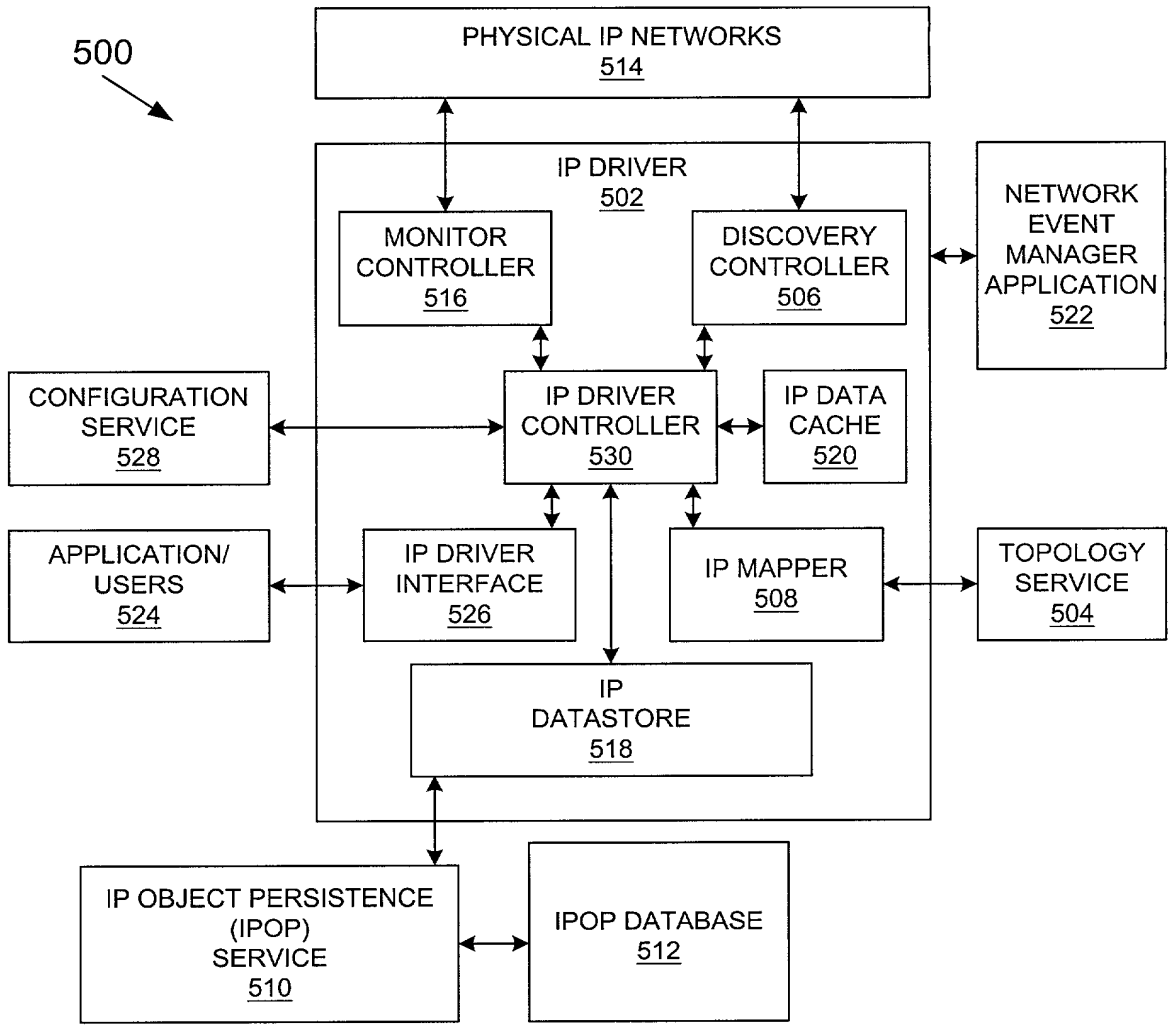
500
PHYSICAL IP NETWORKS
514
IP DRIVER
502
MONITOR CONTROLLER
516
DISCOVERY CONTROLLER
506
NETWORK EVENT MANAGER APPLICATION
522
CONFIGURATION SERVICE
528
IP DRIVER CONTROLLER
530
IP DATA CACHE
520
APPLICATION/ USERS
524
IP DRIVER INTERFACE
526
IP MAPPER
508
TOPOLOGY SERVICE
504
IP DATASTORE
518
IP OBJECT PERSISTENCE (IPOP) SERVICE
510
IPOP DATABASE
512

FIG. 5A

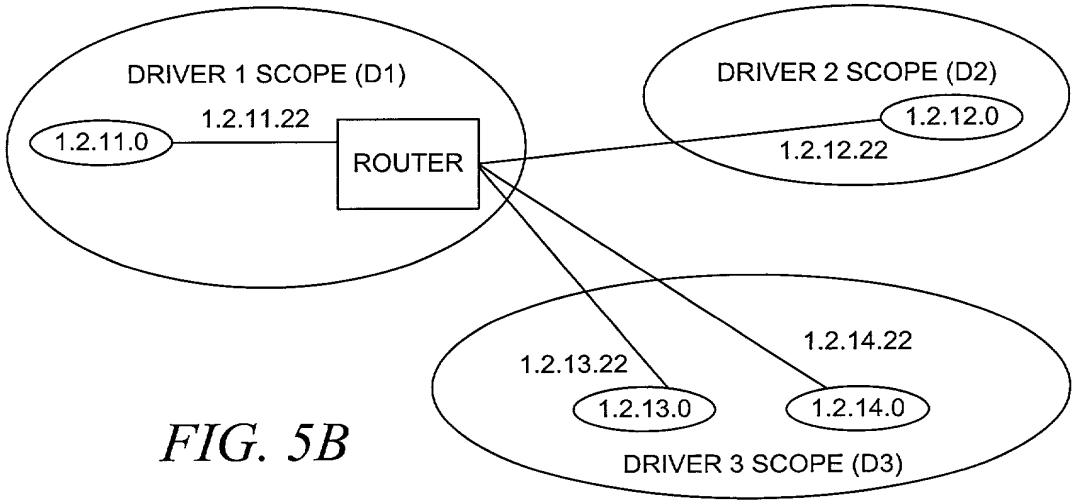
DRIVER 1 SCOPE (D1)
1.2.11.0
1.2.11.22
ROUTER
DRIVER 2 SCOPE (D2)
1.2.12.0
1.2.12.22
1.2.13.22
1.2.14.22
1.2.13.0
1.2.14.0
DRIVER 3 SCOPE (D3)

FIG. 5B

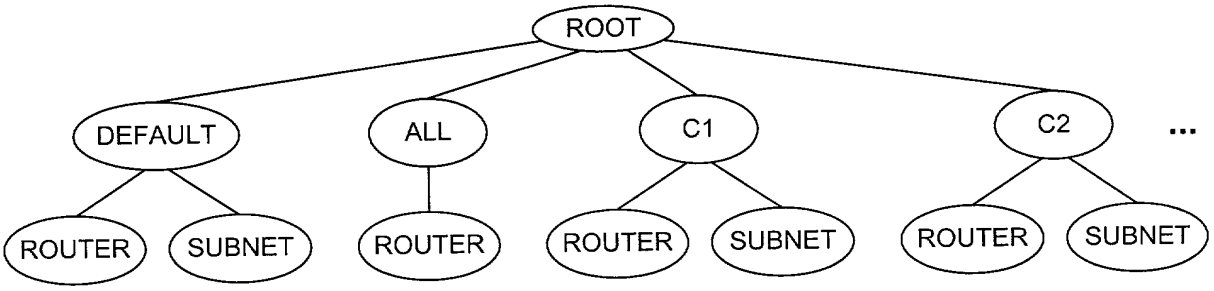
ROOT
DEFAULT
ALL
C1
C2
...
ROUTER
SUBNET
ROUTER
ROUTER
SUBNET
ROUTER
SUBNET

FIG. 5C

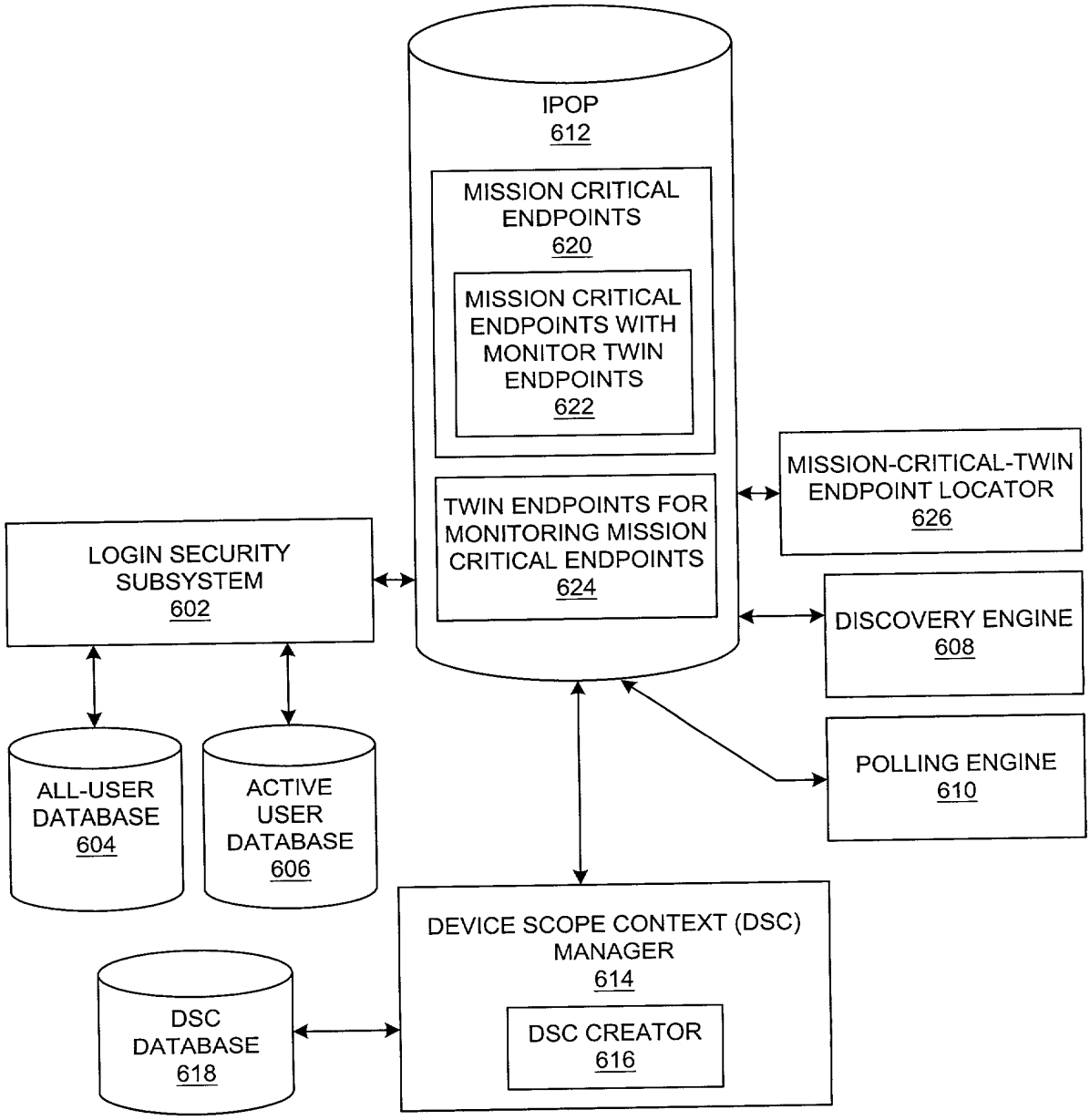
IPOP
612
MISSION CRITICAL ENDPOINTS
620
MISSION CRITICAL ENDPOINTS WITH MONITOR TWIN ENDPOINTS
622
TWIN ENDPOINTS FOR MONITORING MISSION CRITICAL ENDPOINTS
624
MISSION-CRITICAL-TWIN ENDPOINT LOCATOR
626
LOGIN SECURITY SUBSYSTEM
602
DISCOVERY ENGINE
608
POLLING ENGINE
610
ALL-USER DATABASE
604
ACTIVE USER DATABASE
606
DEVICE SCOPE CONTEXT (DSC) MANAGER
614
DSC CREATOR
616
DSC DATABASE
618

FIG. 6

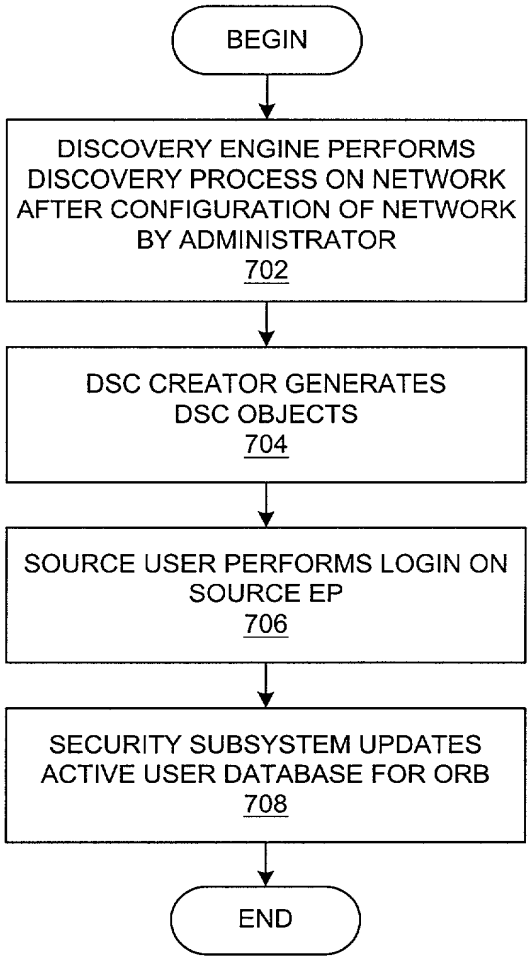
BEGIN
DISCOVERY ENGINE PERFORMS DISCOVERY PROCESS ON NETWORK AFTER CONFIGURATION OF NETWORK BY ADMINISTRATOR
702
DSC CREATOR GENERATES DSC OBJECTS
704
SOURCE USER PERFORMS LOGIN ON SOURCE EP
706
SECURITY SUBSYSTEM UPDATES ACTIVE USER DATABASE FOR ORB
708
END

FIG. 7A

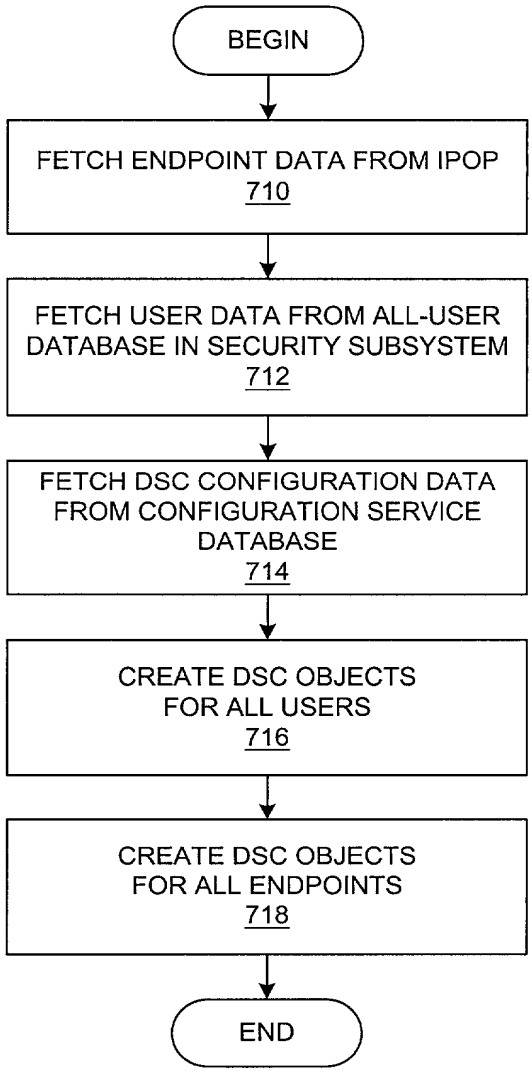
BEGIN
FETCH ENDPOINT DATA FROM IPOP
710
FETCH USER DATA FROM ALL-USER DATABASE IN SECURITY SUBSYSTEM
712
FETCH DSC CONFIGURATION DATA FROM CONFIGURATION SERVICE DATABASE
714
CREATE DSC OBJECTS FOR ALL USERS
716
CREATE DSC OBJECTS FOR ALL ENDPOINTS
718
END

FIG. 7B

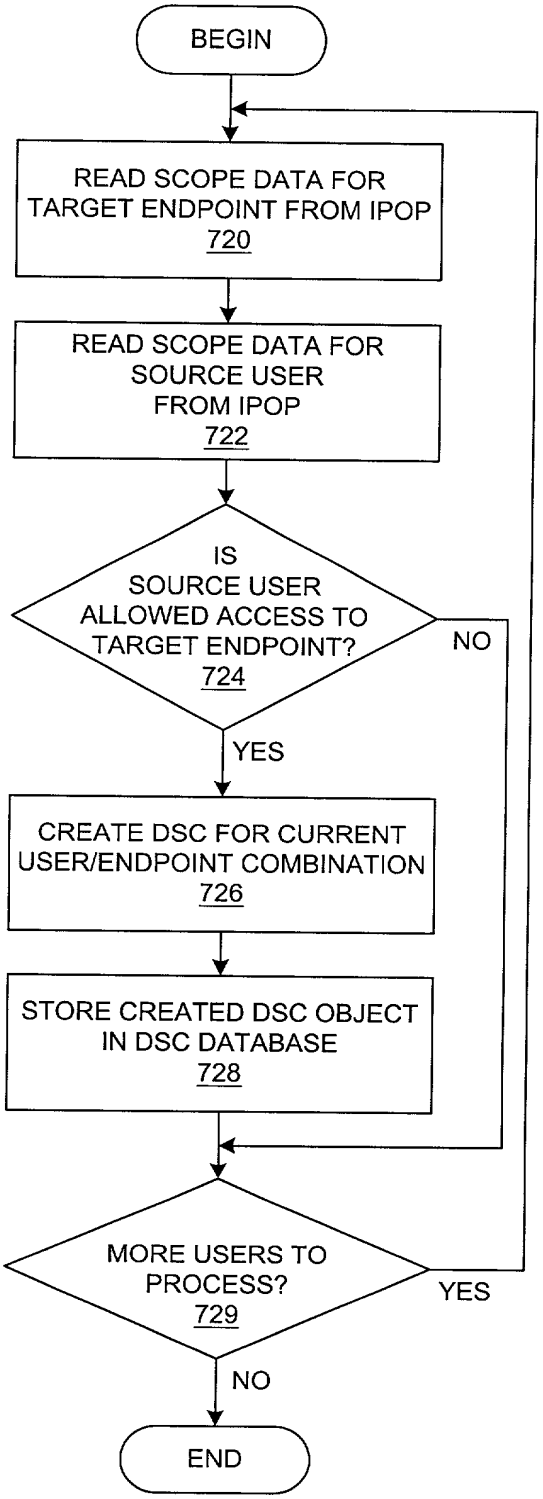
BEGIN
READ SCOPE DATA FOR TARGET ENDPOINT FROM IPOP
720
READ SCOPE DATA FOR SOURCE USER FROM IPOP
722
IS SOURCE USER ALLOWED ACCESS TO TARGET ENDPOINT?
724
NO
YES
CREATE DSC FOR CURRENT USER/ENDPOINT COMBINATION
726
STORE CREATED DSC OBJECT IN DSC DATABASE
728
MORE USERS TO PROCESS?
729
YES
NO
END

FIG. 7C

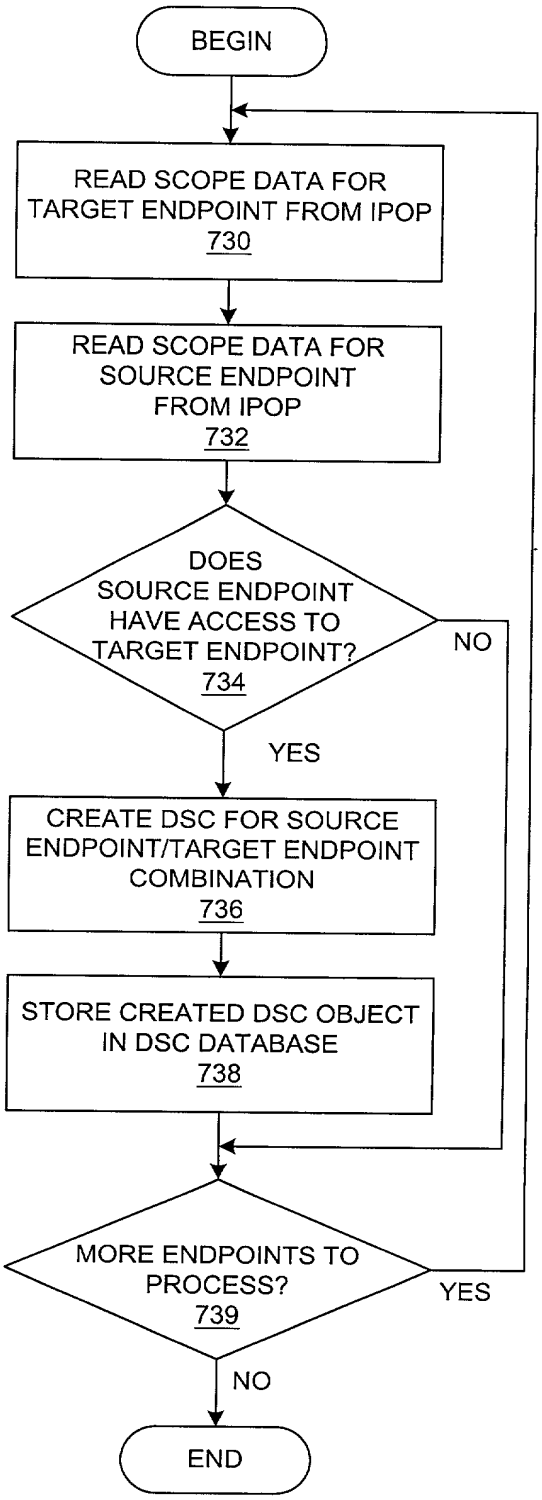
BEGIN
READ SCOPE DATA FOR TARGET ENDPOINT FROM IPOP
730
READ SCOPE DATA FOR SOURCE ENDPOINT FROM IPOP
732
DOES SOURCE ENDPOINT HAVE ACCESS TO TARGET ENDPOINT?
734
NO
YES
CREATE DSC FOR SOURCE ENDPOINT/TARGET ENDPOINT COMBINATION
736
STORE CREATED DSC OBJECT IN DSC DATABASE
738
MORE ENDPOINTS TO PROCESS?
739
YES
NO
END

FIG. 7D

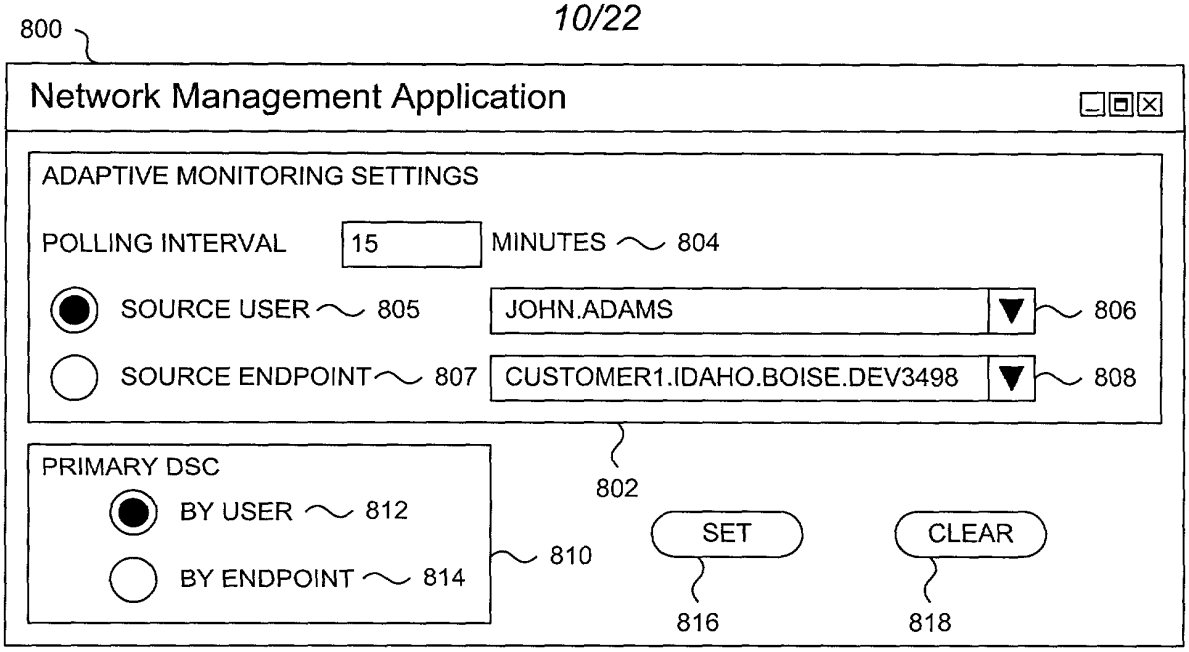

FIG. 8A

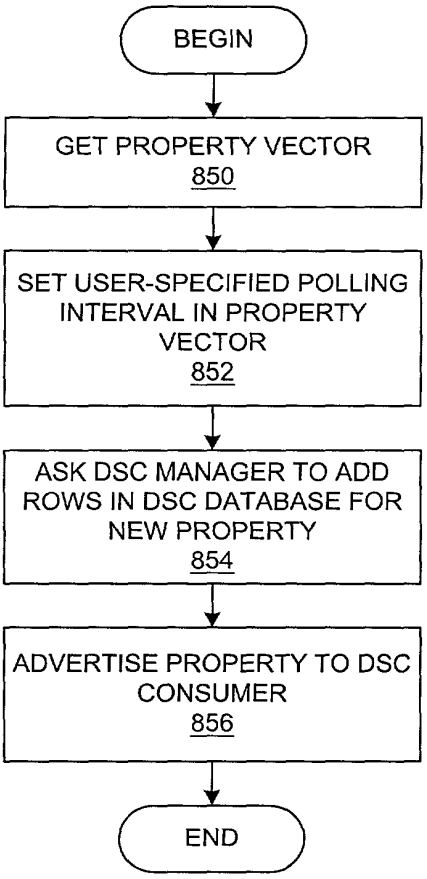

FIG. 8C

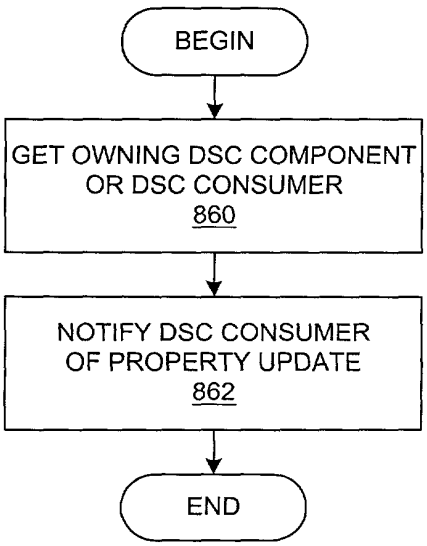

FIG. 8D

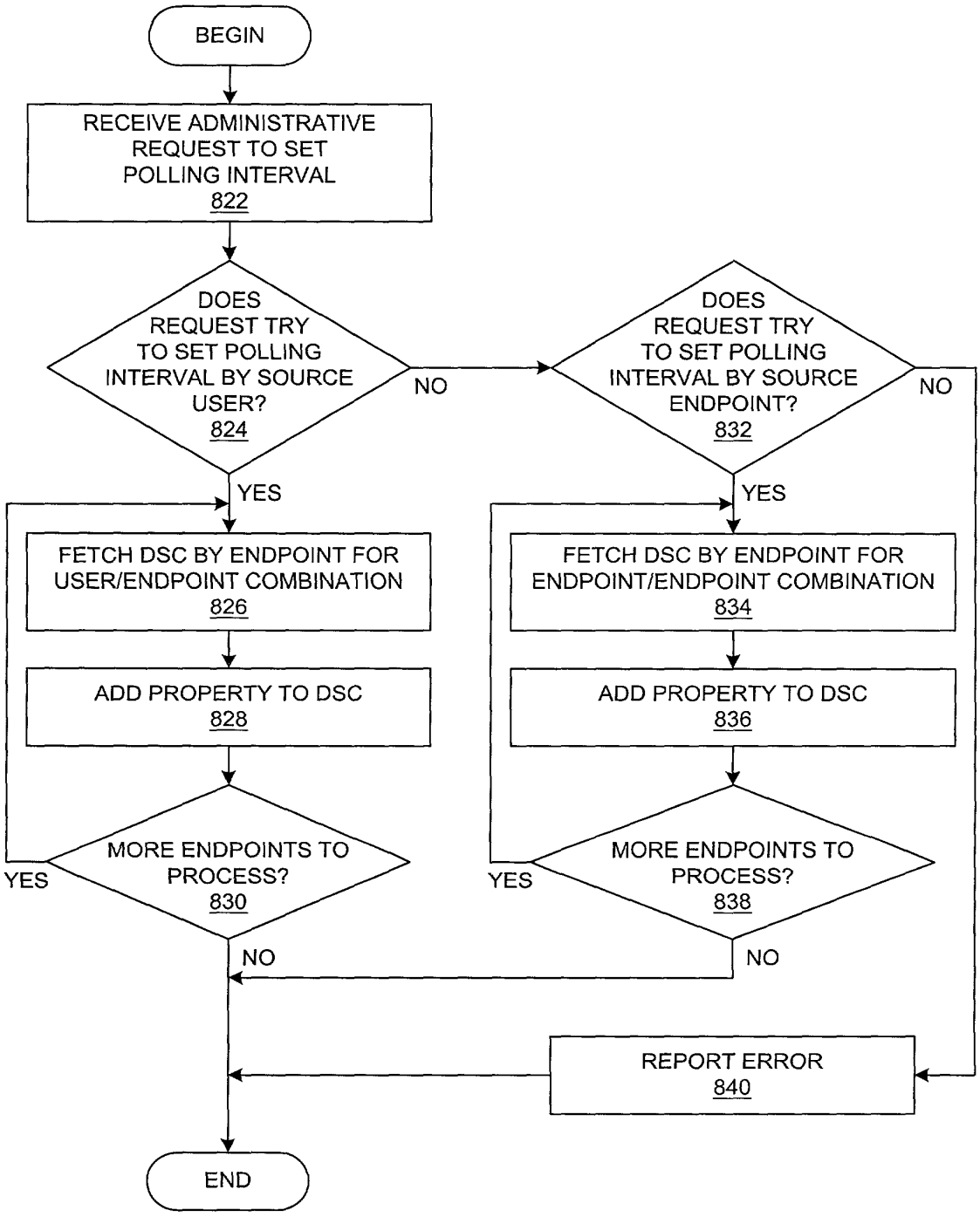
BEGIN
RECEIVE ADMINISTRATIVE REQUEST TO SET POLLING INTERVAL 822
DOES REQUEST TRY TO SET POLLING INTERVAL BY SOURCE USER? 824
NO
YES
DOES REQUEST TRY TO SET POLLING INTERVAL BY SOURCE ENDPOINT? 832
NO
YES
FETCH DSC BY ENDPOINT FOR USER/ENDPOINT COMBINATION 826
ADD PROPERTY TO DSC 828
MORE ENDPOINTS TO PROCESS? 830
YES
NO
FETCH DSC BY ENDPOINT FOR ENDPOINT/ENDPOINT COMBINATION 834
ADD PROPERTY TO DSC 836
MORE ENDPOINTS TO PROCESS? 838
YES
NO
REPORT ERROR 840
END

FIG. 8B

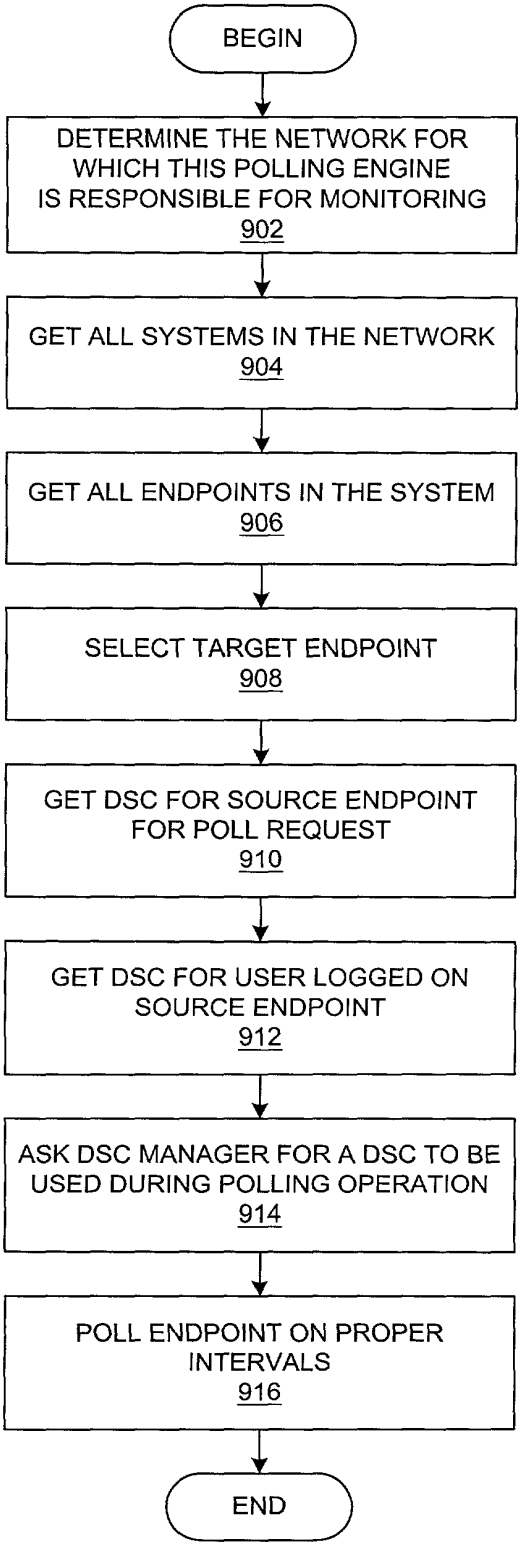
BEGIN
DETERMINE THE NETWORK FOR WHICH THIS POLLING ENGINE IS RESPONSIBLE FOR MONITORING
902
GET ALL SYSTEMS IN THE NETWORK
904
GET ALL ENDPOINTS IN THE SYSTEM
906
SELECT TARGET ENDPOINT
908
GET DSC FOR SOURCE ENDPOINT FOR POLL REQUEST
910
GET DSC FOR USER LOGGED ON SOURCE ENDPOINT
912
ASK DSC MANAGER FOR A DSC TO BE USED DURING POLLING OPERATION
914
POLL ENDPOINT ON PROPER INTERVALS
916
END

FIG. 9A

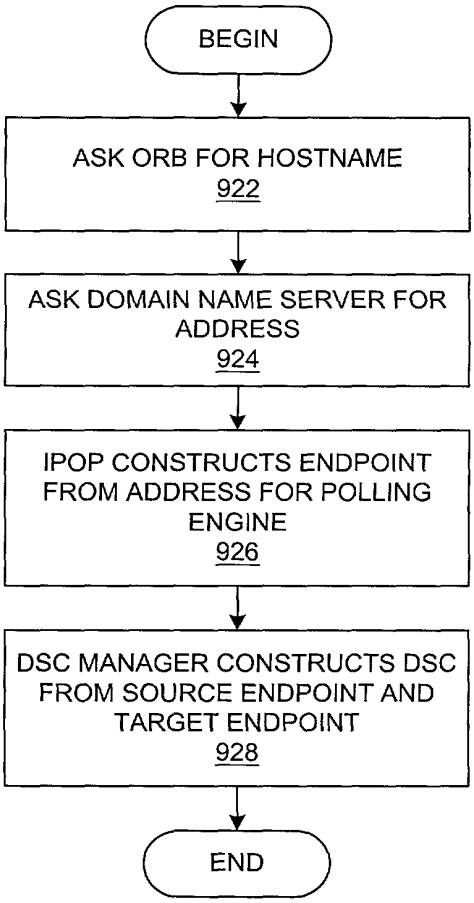
BEGIN
ASK ORB FOR HOSTNAME
922
ASK DOMAIN NAME SERVER FOR ADDRESS
924
IPOP CONSTRUCTS ENDPOINT FROM ADDRESS FOR POLLING ENGINE
926
DSC MANAGER CONSTRUCTS DSC FROM SOURCE ENDPOINT AND TARGET ENDPOINT
928
END

FIG. 9B

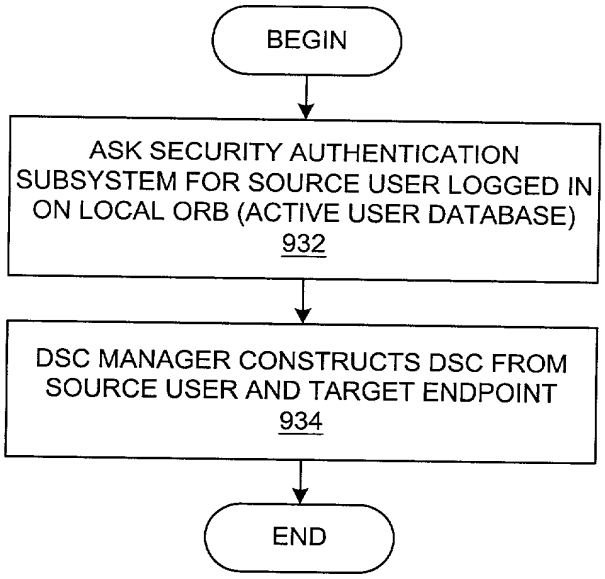

FIG. 9C

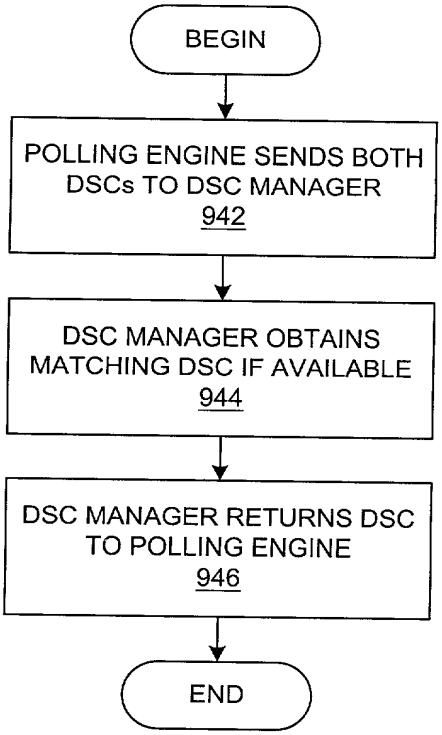

FIG. 9D

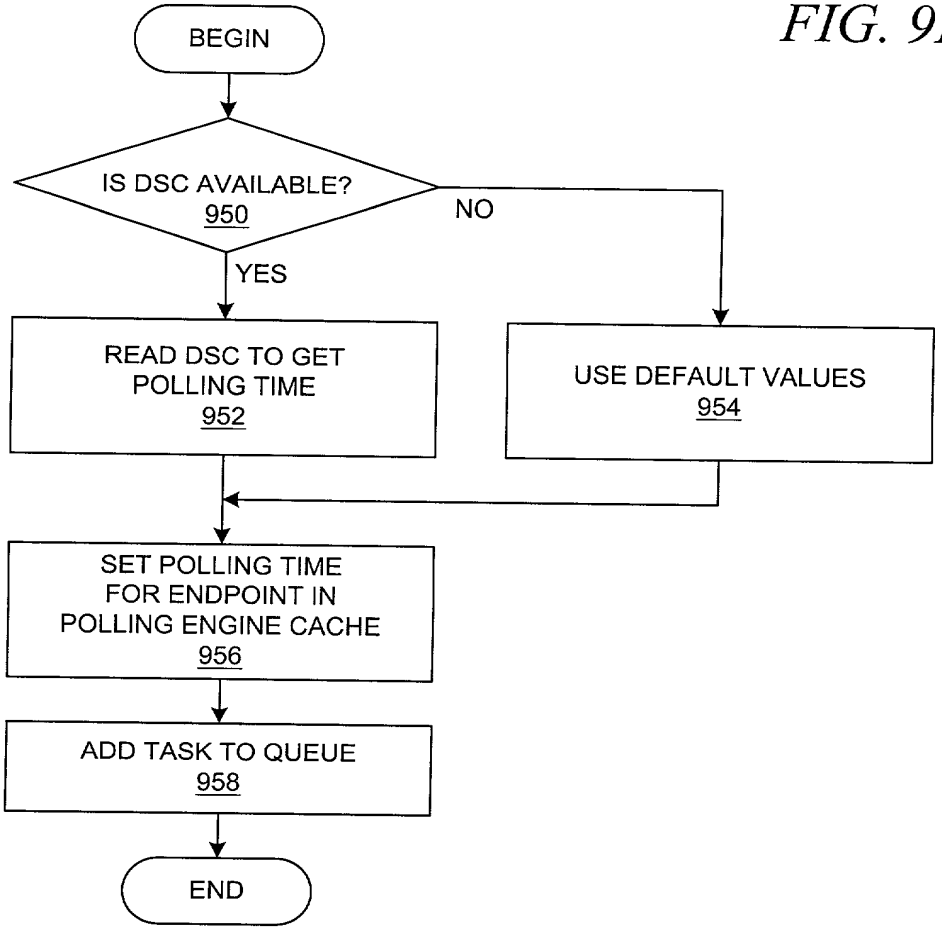

FIG. 9E

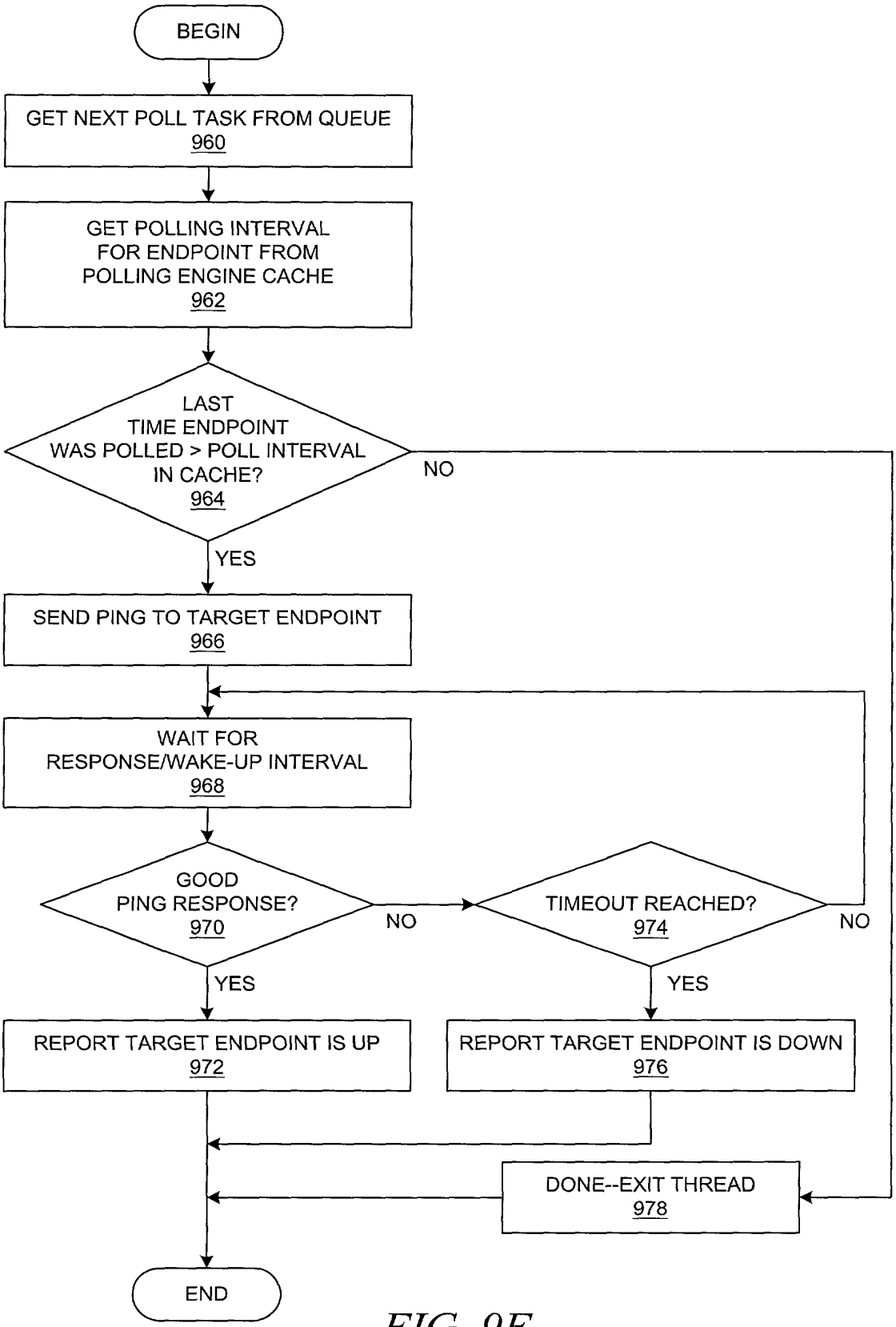
BEGIN
GET NEXT POLL TASK FROM QUEUE
960
GET POLLING INTERVAL FOR ENDPOINT FROM POLLING ENGINE CACHE
962
LAST TIME ENDPOINT WAS POLLED > POLL INTERVAL IN CACHE?
964
NO
YES
SEND PING TO TARGET ENDPOINT
966
WAIT FOR RESPONSE/WAKE-UP INTERVAL
968
GOOD PING RESPONSE?
970
NO
TIMEOUT REACHED?
974
NO
YES
YES
REPORT TARGET ENDPOINT IS UP
972
REPORT TARGET ENDPOINT IS DOWN
976
DONE--EXIT THREAD
978
END

FIG. 9F

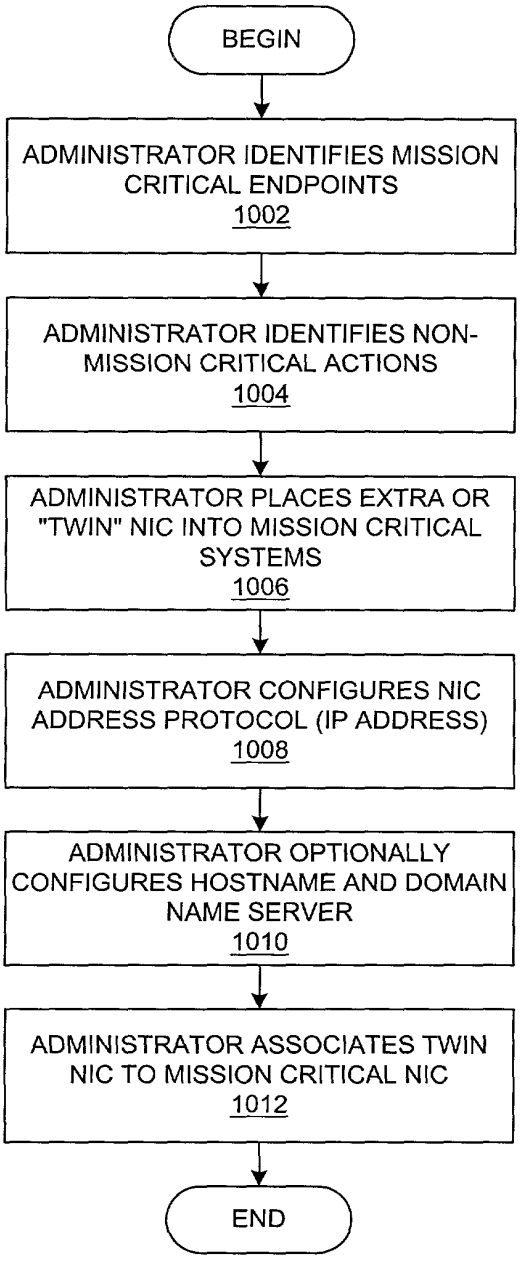

FIG. 10A

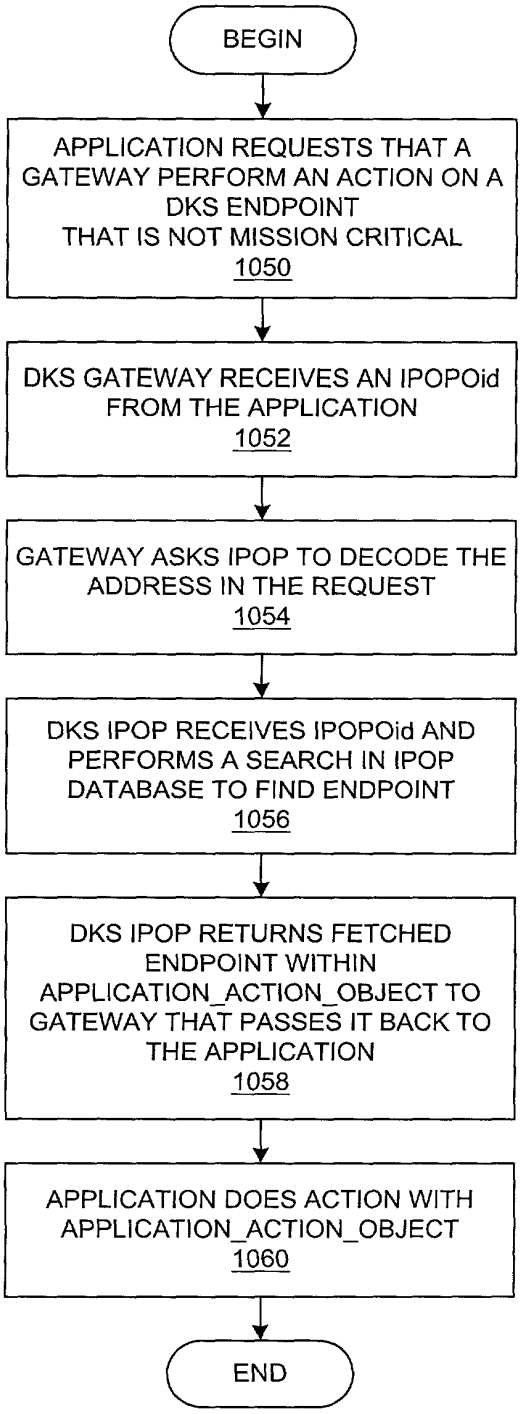

FIG. 10D

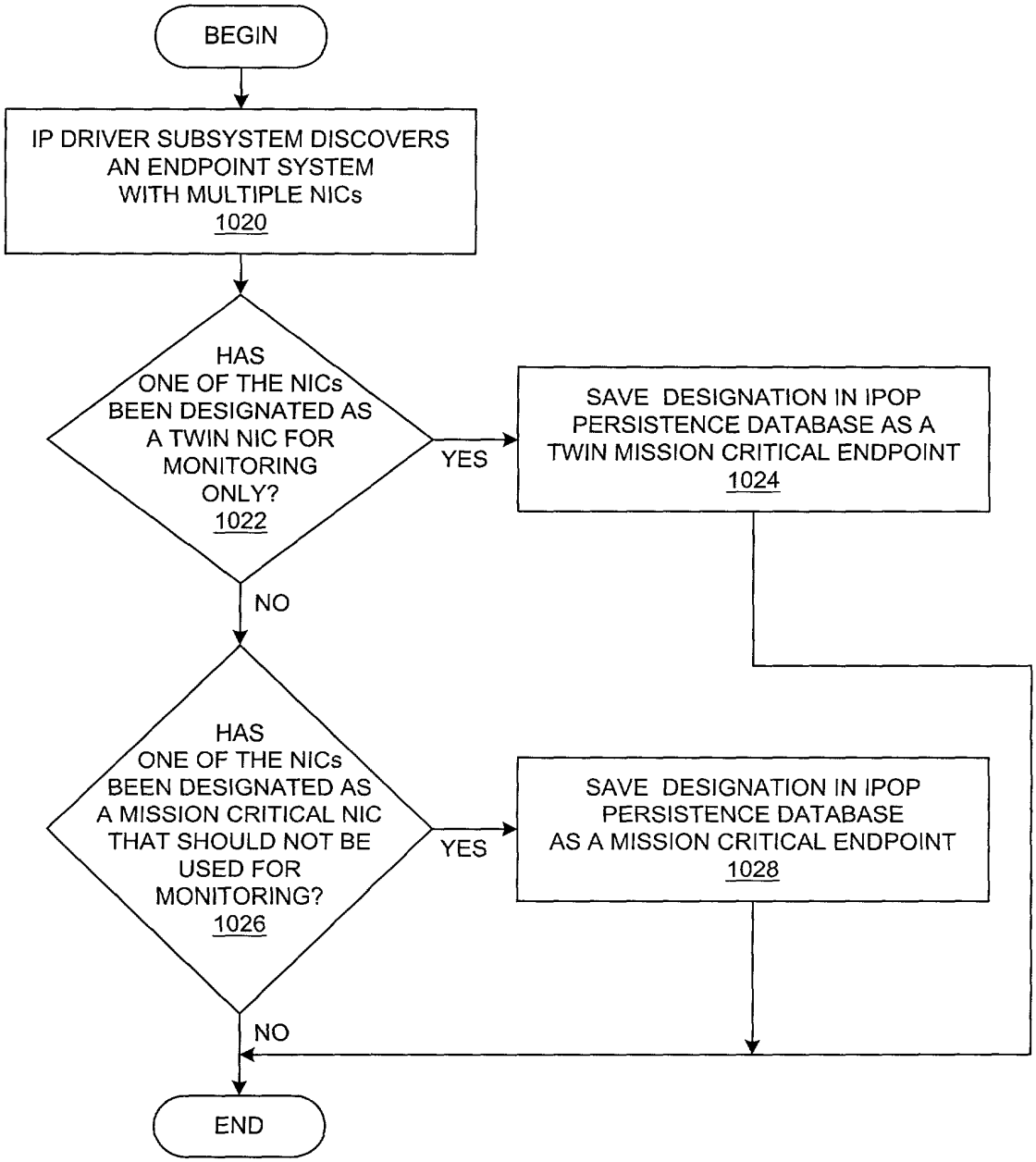
BEGIN
IP DRIVER SUBSYSTEM DISCOVERS AN ENDPOINT SYSTEM WITH MULTIPLE NICs
1020
HAS ONE OF THE NICs BEEN DESIGNATED AS A TWIN NIC FOR MONITORING ONLY?
1022
YES
SAVE DESIGNATION IN IPOP PERSISTENCE DATABASE AS A TWIN MISSION CRITICAL ENDPOINT
1024
NO
HAS ONE OF THE NICs BEEN DESIGNATED AS A MISSION CRITICAL NIC THAT SHOULD NOT BE USED FOR MONITORING?
1026
YES
SAVE DESIGNATION IN IPOP PERSISTENCE DATABASE AS A MISSION CRITICAL ENDPOINT
1028
NO
END

FIG. 10B

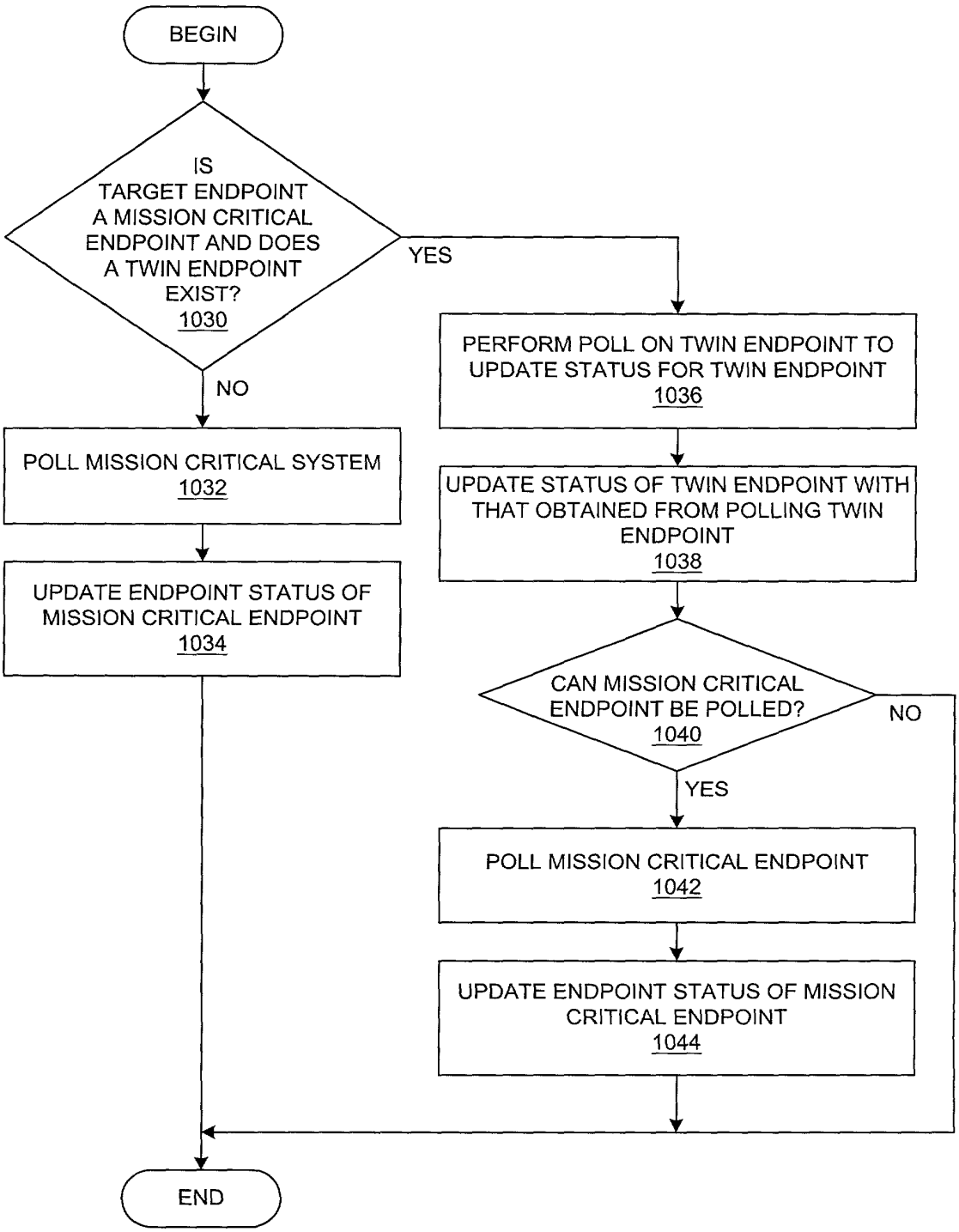
BEGIN
IS TARGET ENDPOINT A MISSION CRITICAL ENDPOINT AND DOES A TWIN ENDPOINT EXIST? 1030
YES
NO
PERFORM POLL ON TWIN ENDPOINT TO UPDATE STATUS FOR TWIN ENDPOINT 1036
UPDATE STATUS OF TWIN ENDPOINT WITH THAT OBTAINED FROM POLLING TWIN ENDPOINT 1038
POLL MISSION CRITICAL SYSTEM 1032
UPDATE ENDPOINT STATUS OF MISSION CRITICAL ENDPOINT 1034
CAN MISSION CRITICAL ENDPOINT BE POLLED? 1040
NO
YES
POLL MISSION CRITICAL ENDPOINT 1042
UPDATE ENDPOINT STATUS OF MISSION CRITICAL ENDPOINT 1044
END

FIG. 10C

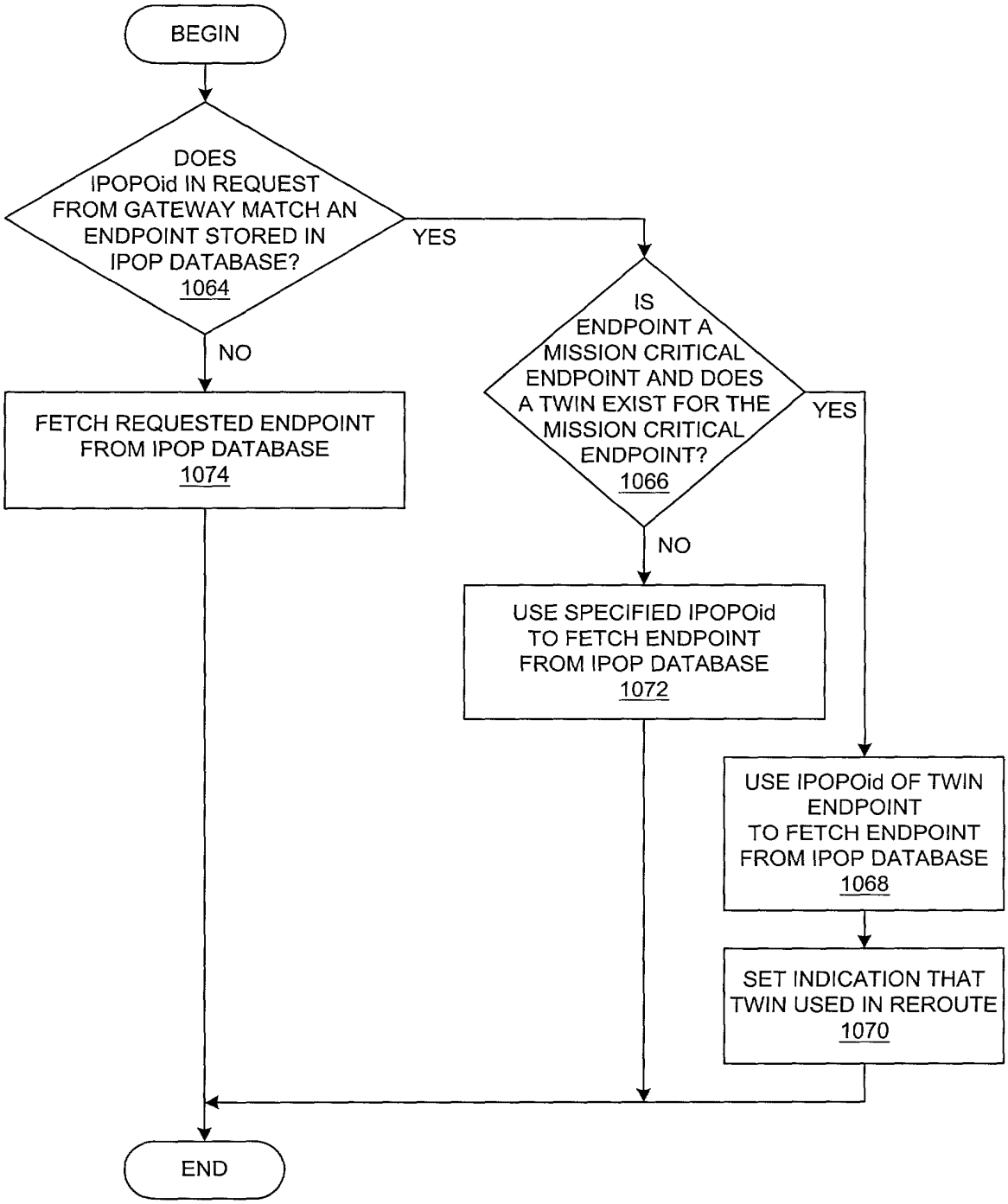
BEGIN
DOES IPOPOid IN REQUEST FROM GATEWAY MATCH AN ENDPOINT STORED IN IPOP DATABASE? 1064
YES
NO
IS ENDPOINT A MISSION CRITICAL ENDPOINT AND DOES A TWIN EXIST FOR THE MISSION CRITICAL ENDPOINT? 1066
YES
NO
FETCH REQUESTED ENDPOINT FROM IPOP DATABASE 1074
USE SPECIFIED IPOPOid TO FETCH ENDPOINT FROM IPOP DATABASE 1072
USE IPOPOid OF TWIN ENDPOINT TO FETCH ENDPOINT FROM IPOP DATABASE 1068
SET INDICATION THAT TWIN USED IN REROUTE 1070
END

FIG. 10E

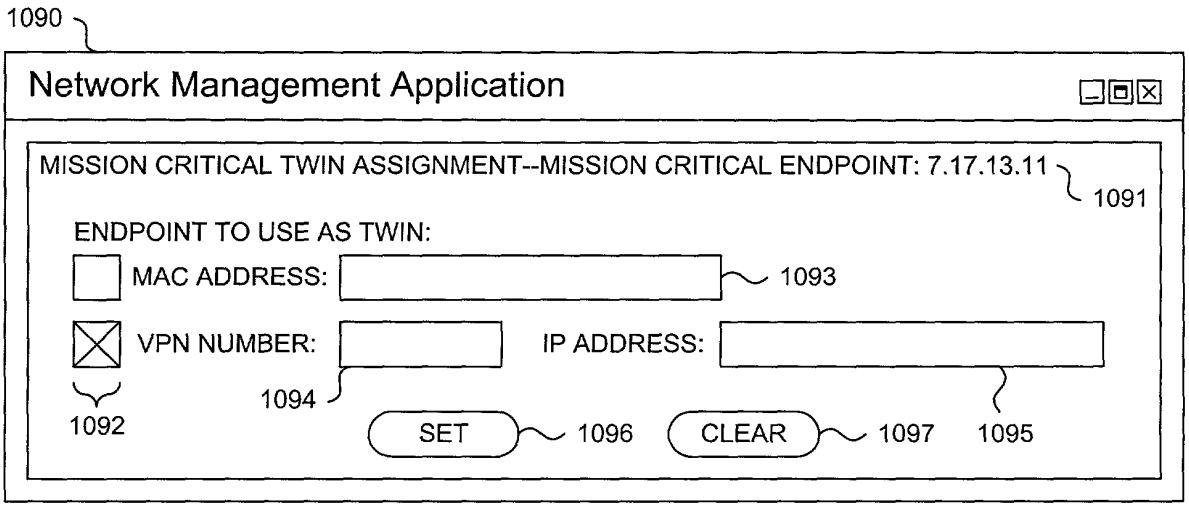

FIG. 10F

```
CLASS  ACTION_OBJECT  {

        // CONSTRUCTOR
        ACTION_OBJECT( LONG IPADDRESS, SHORT VIRTUALPRIVATENETWORKADDRESS )
                THROWS BADADDRESS ...
        .
        .
        .
        VOID PERFORMACTION( ) // EXECUTES ACTION METHOD
.
.
.
}
```

FIG. 11A

```
CLASS  APPLICATION_ACTION_OBJECT  EXTENDS ACTION_OBJECT {

        boolean IsMissionCriticalAction;
                // TRUE = USED TO PERFORM ENTERPRISE-RELATED WORK
                // FALSE = USED TO PERFORM MONITORING OPERATIONS
                //          OR OTHER NON-REVENUE PRODUCING ACTION

        boolean TwinUsedinReroute;
                // TRUE = IPOP HAS REPLACED REQUESTED ADDRESS WITH AN ADDRESS
                //          THAT CAN BE USED FOR NON-MISSION CRITICAL ACTION
.
.
.
}
```

FIG. 11B

```
Public Class Endpoint  {

        //public variables
        long    EPObjectID;  // ID to object (both private and public network addresses)
        InetAddress  EPIPAddress; // physical network address (private or public)
        long    EPVPN;       // virtual private network ID

        // get/set of variables
        public long          getObjectID( ) { ... }
        public InetAddress   getPAddress( ) { ... }
        public long          getVPN( ) { ... }

}
```

FIG. 11C

```
Class TwinMissionCriticalEndpoint extends Endpoint {
.
.
.
        IPOPOid      missionCriticalEndpoint;
                          // Mission critical endpoint that is used to gather status
        IPOPOid      missionCriticalSystem;
                          // Mission critical system

        long         endpointStatus;
        long         twinEndpointStatus;

.
.
.
}
```

FIG. 11D

```
Class MissionCriticalEndpoint extends Endpoint {
.
.
.
boolean      useForMonitoring;  // TRUE = endpoint can be used for monitoring,
                                // polling, and other system-management-type
                                // resources
.
.
.
}
```

FIG. 11E

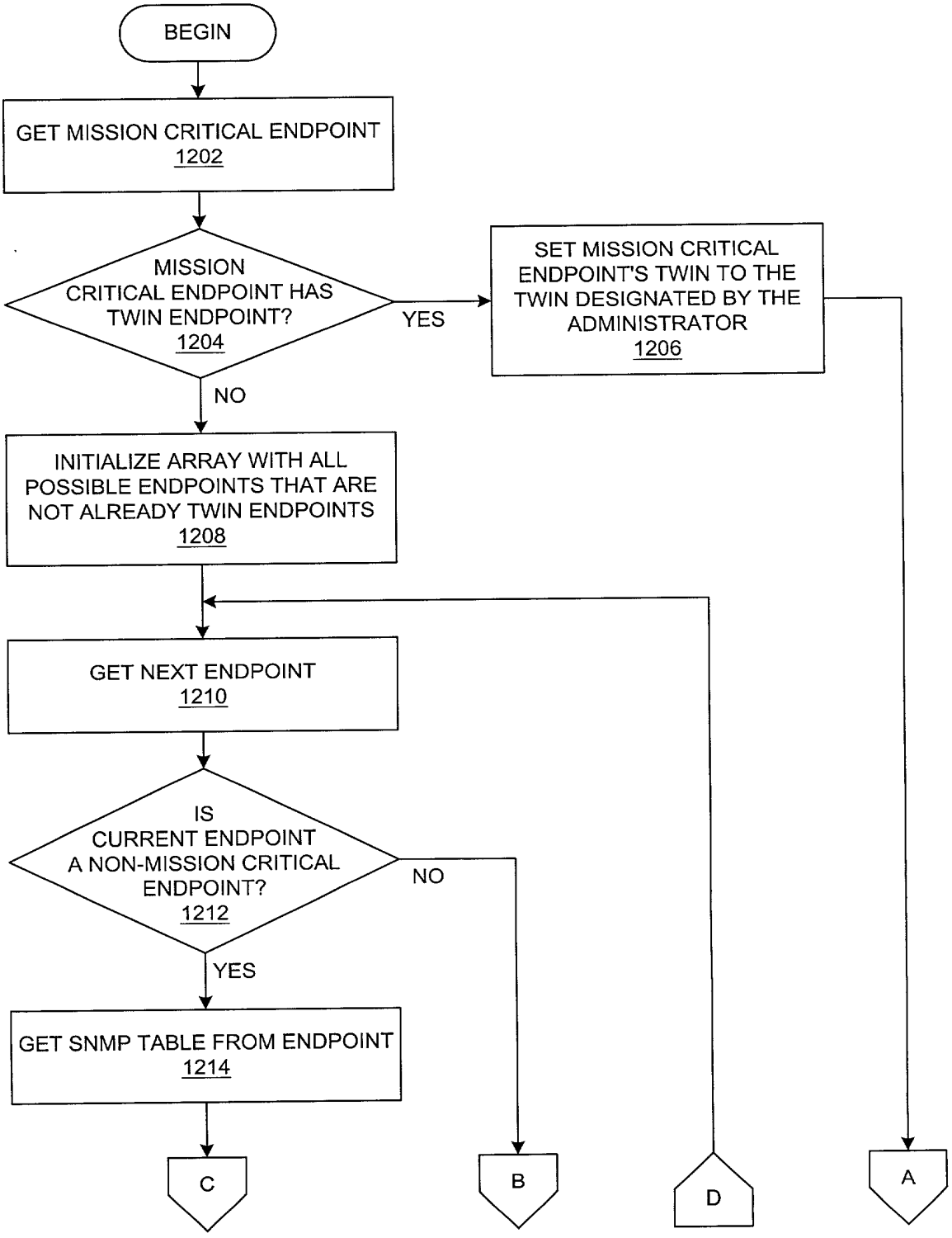
BEGIN
GET MISSION CRITICAL ENDPOINT
1202
MISSION CRITICAL ENDPOINT HAS TWIN ENDPOINT?
1204
YES
SET MISSION CRITICAL ENDPOINT'S TWIN TO THE TWIN DESIGNATED BY THE ADMINISTRATOR
1206
NO
INITIALIZE ARRAY WITH ALL POSSIBLE ENDPOINTS THAT ARE NOT ALREADY TWIN ENDPOINTS
1208
GET NEXT ENDPOINT
1210
IS CURRENT ENDPOINT A NON-MISSION CRITICAL ENDPOINT?
1212
NO
YES
GET SNMP TABLE FROM ENDPOINT
1214
C
B
D
A

FIG. 12A

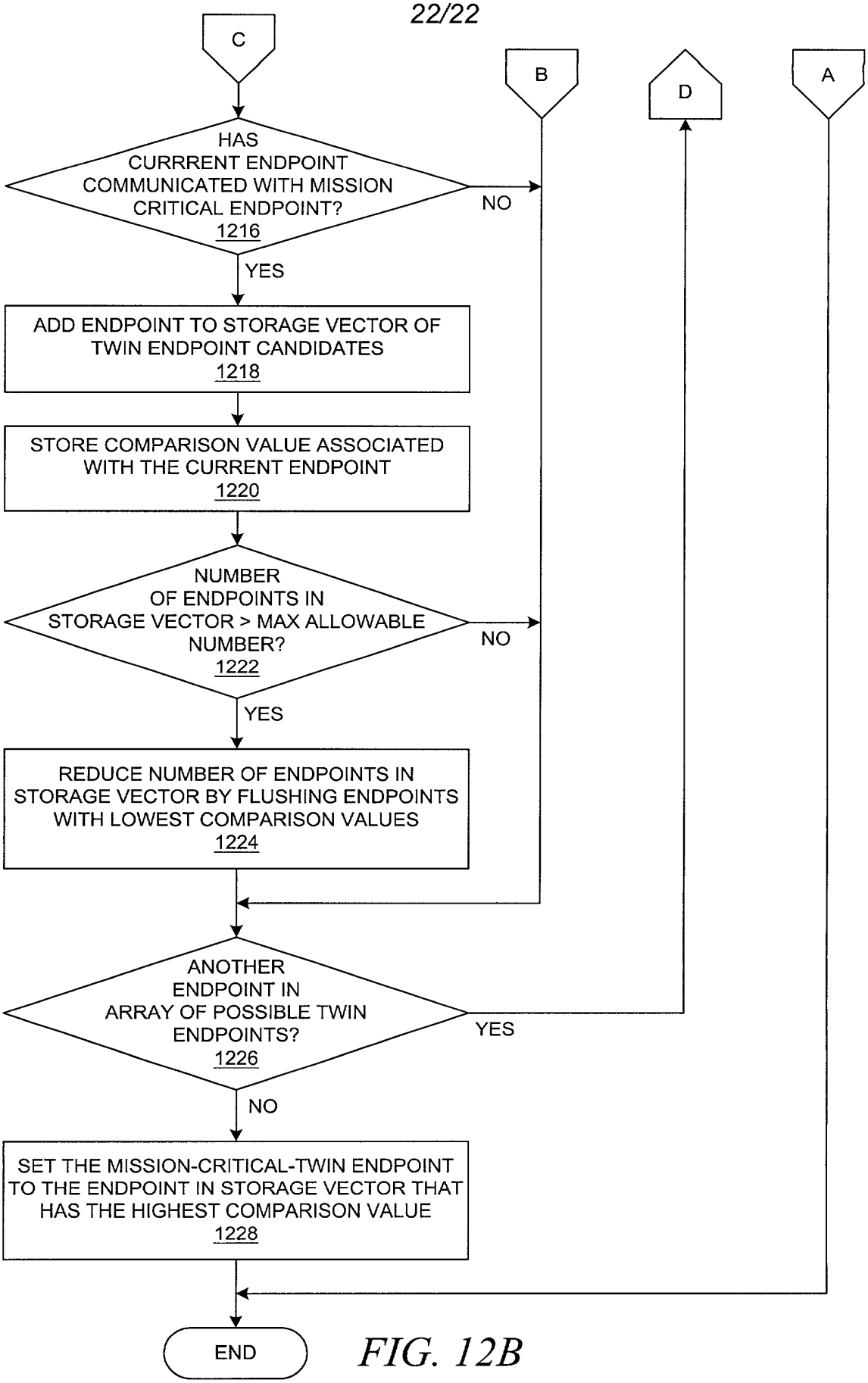
C
B
D
A
HAS CURRRENT ENDPOINT COMMUNICATED WITH MISSION CRITICAL ENDPOINT?
1216
NO
YES
ADD ENDPOINT TO STORAGE VECTOR OF TWIN ENDPOINT CANDIDATES
1218
STORE COMPARISON VALUE ASSOCIATED WITH THE CURRENT ENDPOINT
1220
NUMBER OF ENDPOINTS IN STORAGE VECTOR > MAX ALLOWABLE NUMBER?
1222
NO
YES
REDUCE NUMBER OF ENDPOINTS IN STORAGE VECTOR BY FLUSHING ENDPOINTS WITH LOWEST COMPARISON VALUES
1224
ANOTHER ENDPOINT IN ARRAY OF POSSIBLE TWIN ENDPOINTS?
1226
YES
NO
SET THE MISSION-CRITICAL-TWIN ENDPOINT TO THE ENDPOINT IN STORAGE VECTOR THAT HAS THE HIGHEST COMPARISON VALUE
1228
END

FIG. 12B